

## **METHOD AND APPARATUS FOR LIMITING REUSE OF DOMAIN NAME SYSTEM RESPONSE INFORMATION**

### **Technical Field**

[0001] The present invention relates generally to computer network systems, and more particularly to a method and apparatus for limiting the reuse of domain name system information, such as internet protocol (IP) addresses assigned to a particular domain name, which is periodically validated.

### **Background of the Invention**

[0002] The domain name system (DNS) is an integral part of the Internet and other networks that use Internet-type protocols and architectures, such as transmission control protocol internet protocol (TCP/IP). The primary function of the domain name system is to permit users to access information on different computers connected to the Internet by simply typing or selecting text names, without regard to the sequence of numbers that specify the actual machine addresses or the name servers where the information to be accessed is located.

[0003] Thus, DNS is responsible for mapping an ASCII hostname, such as www.research.att.com to an IP address, such as 192.20.225.10. In some cases, different IP addresses are returned in response to different queries, which may be a function of information, such as load and network location. In addition, a single response can provide multiple IP addresses.

[0004] The entire database that is accessible over the Internet is divided into zones, which are distributed among name servers. The essential task of a name server is to answer queries using data in its zone. The reliability of the system and the information contained therein is of paramount importance. Consequently, and in order to ensure continued reliability, the DNS requires that at least two name servers within a zone contain data derived from an authoritative server, i.e., authoritative data. The two name servers may be denominated primary and secondary servers, or, as referred to herein, higher level DNS servers and lower level DNS servers.

One of the chief responsibilities of name servers is to load the database files and maintain cached data acquired through resource records. Each resource record contains a so-called "time-to-live" field, which ensures that name servers do not cache data without a time bound.

[0005] The DNS is a hierarchical service, in which one DNS server can obtain information from another and then reuse that information repeatedly until a time-based threshold is reached. The time-based threshold is referred to as "time-to-live" or TTL. Normally, a local DNS server acts on behalf of clients by requesting DNS records, caching them, and then serving them to other clients for a period of time, known as the TTL, without revalidating the information. When an Internet user makes a request for a domain name record or information from a DNS server within the DNS, the DNS server must obtain the requested information or record and send it to the requesting party.

[0006] One method of accomplishing this task is for the server to check its cache memory for a local list of domain names already matched to machine addresses. If the requested domain name is in cache, then the computer can respond with an answer in the form of the associated machine address within a very short period of time. Such a local list is referred to as a name cache. It is generally stored in random access memory, disk storage, or another storage device coupled to the name server computer. The name cache is updated periodically from other name servers in the Internet.

[0007] When a DNS server caches the data, it builds up a database with relevant information that may be useful the next time the same or a similar query is received. Essentially, once the information is cached, the name server acquires significant knowledge about the data and whether it is responsive to a particular query or class of queries. Thus, once the information is cached, the process of obtaining the data is shortened significantly because the server now has the answer to the query in its cache. This leads to greater efficiency and enhanced performance, insofar as the cached information can be used for future inquiries. The resolution of data that is already in the local cache is considerably more efficient than a remote resolution that requires maneuvering through network traffic. However, there is a tradeoff. There is

a very real possibility that the cached information may become outdated by the time it is used.

[0008] The local DNS server may serve a small number of clients, or an enormous community. Until the TTL expires, the local DNS will continue to use and distribute the same information. On the one hand, with a short TTL, the information will be fairly current, even with many clients using the specified host, because a new host can be selected before the load peaks too dramatically.

[0009] However, revalidating the information and establishing a new TTL requires that queries be sent to the authoritative server linked to the local DNS server. Thus, while the information will be current, some of these clients must wait for a round-trip between the local DNS and the authoritative DNS. On the other hand, with a longer TTL, the DNS lookup is amortized over more lookups, and fewer clients must wait, but the server may become overloaded before a new host is used.

[0010] Originally, DNS was intended to be static, that is, a hostname-to-IP address assignment would rarely change and would certainly not be different for different hosts requesting the same information. With the advent of DNS being used for geographic load balancing, DNS is much more dynamic, but there is still a balance to be drawn between assigning a low TTL to ensure that too many hosts do not use the information before revalidating, and assigning a higher TTL to limit the delay in repeatedly obtaining the same DNS information.

[0011] As discussed above, DNS is a hierarchical service, in which one DNS server can obtain information from another and then reuse that information until the TTL has expired. There is currently no control over whether the DNS server reuses this information to serve a small number of requests or a large number of requests. This results in what could be called a "hidden load" that can only be estimated using prior knowledge concerning a particular DNS server. For instance, a server associated with a high rate of usage would, if possible, be assigned a shorter TTL and/or directed to a host with greater capacity than a server having a more moderate rate of usage. In view of the foregoing considerations, a method and system are called for to address this hidden load problem.

**Summary of the Invention**

[0012] It is an object of the present invention to provide a system and method that limit use of specific domain name system (DNS) information according to a quantity of clients and servers requesting this information.

[0013] It is another object of the present invention to provide a system and method that pass information concerning the quantity of clients and servers that have requested specific DNS information to a server that has been requested to provide this information.

[0014] A method of limiting reuse of domain name information in accordance with one form of the present invention, which incorporates some of the preferred features, includes the step of requesting domain name information by a local domain name server from an authoritative domain name server. The domain name information includes an allowable usage limit that represents a maximum number of times that the domain name information can be provided to a requestor, such as a client or server, before validating the domain name information.

[0015] A system for limiting reuse of domain name information formed in accordance with one form of the present invention, which incorporates some of the preferred features, includes an authoritative domain name server, a local domain name server, and at least one requestor, such as a client or server. The local domain name server is operatively coupled to the authoritative domain name server, and the requestor is operatively couple to the local domain name server.

[0016] The requestor requests domain name information from the local domain name server, and the local domain name server requests domain name information from the authoritative domain name server. The authoritative domain name server provides the domain name information in response to the local domain name server requesting domain name information. The domain name information includes an allowable usage limit, which represents a maximum number of times the domain name information can be provided to the requestor before validating the domain name information.

[0017] Other objects and features of the present invention will become apparent from the following detailed description considered in conjunction with the accompanying drawings. It is to be understood, however, that the drawings are designed as an illustration only and not as a definition of the limits of the invention.

### **Brief Description of the Drawings**

[0018] Figure 1 is a block diagram of a system for limiting reuse of domain name information formed in accordance with the present invention.

[0019] Figure 2a is a flowchart of a first embodiment of a method of limiting reuse of domain name information in accordance with the present invention.

[0020] Figure 2b is a flowchart of a second embodiment of the method of limiting reuse of domain name information in accordance with the present invention.

[0021] Figure 3a is a flowchart of a third embodiment of the method of limiting reuse of domain name information in accordance with the present invention.

[0022] Figure 3b is a flowchart of a fourth embodiment of the method of limiting reuse of domain name information in accordance with the present invention.

### **Detailed Description**

[0023] Domain name system (DNS) information is generally requested from an authoritative DNS server by a local DNS server. The authoritative server responds by providing the DNS information to the local DNS server. The system and method in accordance with the present invention limits reuse of domain name information by returning the requested DNS information with an allowable usage limit, in addition to the customary time-to-live (TTL) threshold. The allowable usage limit represents the maximum number of times that the associated domain name information may be provided to requestors before it must be revalidated.

[0024] The local DNS server returning the requested DNS information to a client preferably decrements a residual usage counter, which is a countdown of the allowable usage limit. The local DNS server that returns the requested DNS information to another DNS server would preferably issue a subset of its own

allowable usage limit to the requesting server, and decrement its own residual usage counter by that portion of the allowable usage limit granted to the requesting server to ensure that the allowable usage limit remains constant.

[0025] In this way, the local DNS server preferably enables another DNS server to permit a maximum number of clients to use a particular IP address before revalidating. When the server at the requested IP address is lightly loaded, the number of uses and the duration of the TTL in the response are preferably relatively high. In contrast, when the server or other parts of the infrastructure, such as network or load-balancing switches, approach maximum capacity, the TTL is preferably more conservative.

[0026] Another embodiment in accordance with the present invention enables the DNS server requesting revalidation of DNS information to provide feedback information concerning the number times that the DNS information has been provided to other clients and servers as an observed usage count. This information is preferably used to improve the assignment of IP addresses to achieve improved load balancing of the network.

[0027] In the case of a plurality of IP addresses being assigned to one hostname, the same or different allowable usage limits may be assigned to the IP addresses associated with a particular hostname. The DNS server requesting the DNS information could then assign the IP addresses to other servers or clients randomly, sequentially (round robin), or probabilistically in accordance with weights assigned to each of the plurality of IP addresses. For example, in a probabilistic assignment, assume that there are two IP addresses, IPA1 and IPA2, assigned to a hostname. Further assume that IPA1 is assigned a weighting of 0.4 (40%) and that IPA2 is assigned a weighting of 0.6 (60%). The DNS server requesting the DNS information could then assign IPA1 to 40% of its requestors and assign IPA2 to 60% of its requestors.

[0028] Figure 1 shows a system 10 for limiting reuse of DNS information. The system includes an authoritative DNS server 12 operatively coupled to a local DNS server 14. The local DNS server 14 is operatively coupled to other local DNS servers 18, 21, 22. The system also includes at least one client 16 that is operatively

coupled to and requests DNS information from the local DNS server 14. The clients 16 preferably access a web server 20 using DNS information obtained from the authoritative DNS server 12, and one or more of the local DNS servers 14, 18, 21, 22. The authoritative DNS server 12, which returns an A record or DNS information including a hostname-to-IP-address mapping, may realize that some local DNS servers tend to serve a large number of client 16s, and consequently may direct those servers to point their clients 16 at particular hosts that have unusually large capacity, and/or return a shorter than usual TTL.

[0029] Figure 2a is a flowchart of a first embodiment of a method of limiting reuse of domain name information in accordance with the present invention. In using the method, a client requests DNS information in the form of a record from the local DNS server. Upon receiving the DNS information request from the client in step 24, the local DNS server preferably determines whether the requested DNS information is in its cache in step 26. If the requested information is in cache, the local DNS server then determines whether the cached DNS information has expired in step 28, that is, the server determines whether the time to live (TTL) threshold associated with the DNS information is zero.

[0030] If the local DNS server determines that the cached DNS information has not expired in step 28, that is, the TTL value is not zero, the local DNS server determines the value of a residual usage counter in step 30. If the value of the residual usage counter is not zero, the local DNS server decrements the value of the residual usage counter in step 32, and provides the requested DNS information to the client that initially requested the information in step 34.

[0031] Alternatively, the local DNS server may determine that the requested DNS information is not cached in step 26, or if it is cached, that the cached DNS information has expired in step 28, or that although the TTL has not expired ( $TTL > 0$ ), the residual usage counter value is 0. In each of these circumstances, the local DNS server preferably requests and obtains the DNS information from the authoritative DNS server, which includes a new TTL and allowable usage limit N in step 36.

[0032] By obtaining the new TTL and allowable usage limit, the authoritative DNS server essentially eliminates the conditions that caused the local DNS server to request the information from the authoritative server. The TTL reflects the amount of time that the DNS information may exist before being revalidated. The residual usage counter value preferably reflects the number of times that the local DNS server may provide the domain name information to a requester, such as a client or another server.

[0033] Upon receipt of the requested DNS information from the authoritative DNS server, the local DNS server preferably stores the information in its local cache in step 38. The local DNS server then preferably sets the residual usage counter to N-1 in step 40, and sends the requested DNS information to the client in step 34. By setting the residual usage counter to N-1, the local DNS server decrements the residual usage count by one to account for the DNS information being providing to the requesting client for the first time in step 34.

[0034] Figure 2b represents a second embodiment of the method of limiting reuse of domain name information in accordance with the present invention, which is substantially similar to that shown in Figure 2a, except that the DNS information is being requested by and provided to a server rather than a client. Specifically, in the embodiment shown in Figure 2b, the initial request for DNS information preferably originates from a local DNS server to another local DNS server, such as from server 18, 21 to server 14 or from server 22 to server 18 shown in Figure 1, rather than from the client 16 to the local DNS server 14.

[0035] As described above in reference to Figure 2a, when the DNS server receives the request in step 42, the server preferably determines whether the requested information is already in its cache memory in step 44 of Figure 2b. If the requested DNS information exists in cache, the DNS server determines that the TTL has not expired in step 46, and the residual usage counter value is not zero in step 48, then the DNS server preferably decrements the residual usage counter in step 50, and sends the requested information to the requesting DNS server in step 52.

[0036] However, in providing the DNS information, the DNS server also preferably sends an allowable usage limit subset M in step 52 associated with the



requested information, which represents a portion of the allowable usage limit originally provided with the DNS information by the authoritative DNS server when first validated. The allowable usage limit M preferably reflects the number of requesters, which include clients and servers, to which the requesting DNS server may subsequently provide the corresponding information. Accordingly, the residual usage counter is preferably decremented by M in step 50 to represent that portion of the allowable usage counter being granted to the requesting DNS server in step 52.

[0037] In Figure 2b, if the requested information does not exist in cache, and the local DNS server must obtain it from the authoritative server in step 44, the authoritative server sets the allowable usage limit N associated with the DNS information before sending it and the requested information to the local DNS server in step 54. Upon receiving the information from the authoritative server, the local DNS server preferably stores the information in cache in step 56, sets the residual usage counter to N-M in step 58, and sends the requested DNS information to the requesting DNS server with an allowable usage limit subset M, which represents a portion of the allowable usage limit N, in step 52. The requesting DNS server, which has received the requested information and the associated allowable usage limit subset M, preferably decrements M each time this information is provided to a requestor by an amount equal to the usage limit granted to that requestor.

[0038] Figure 3a shows a third embodiment of the method of limiting reuse of domain name information in accordance with the present invention. In this embodiment, when the local DNS server receives a request for DNS information from the client in step 60 and determines that the requested information is not in its cache in step 62, the server preferably initializes an observed usage counter associated with the requested information to zero in step 74.

[0039] The server then preferably sends a request for the DNS information to the authoritative DNS server with the current value of the observed usage counter in step 76. In response, the authoritative DNS server preferably provides the requested DNS information with the associated TTL value and the allowable usage limit N to the requesting client in step 78. Thus, the requesting DNS server essentially obtains information from the authoritative server that is defined by a set of three parameters, namely, TTL and the allowable usage limit N provided by the

authoritative server, as well as the observed usage counter provided by the requesting server.

[0040] Thereafter, the DNS server preferably stores the information in its local cache in step 80 so that it is readily available for the next client requesting the same information. In storing the information in cache, the local DNS server preferably sets the associated residual usage counter to  $N-1$  in step 82 to account for the fact that it is about to provide the information to the requesting client. The server also preferably increments the observed usage counter in step 84 to represent that the information is being provided to the requesting client. Thus, for every decrement in the residual usage counter, there is preferably an accompanying increment in the observed usage counter, and both figures represent the fact that the requested information is being provided to the requesting client in step 72.

[0041] If the requested DNS information is found to be in cache in step 62, and either the DNS information has expired in step 64, or the associated residual usage counter is zero in step 66, the server preferably requests the DNS information from the authoritative server and provides the current observed usage counter value to the authoritative server in step 76. If the requested DNS information is found to be in cache in step 62, the DNS information has not expired in step 64, and the associated residual usage counter is not zero in step 66, the observed usage counter is incremented in step 68 and the residual usage counter is decremented in step 70 to represent that the DNS information is being provided to the requesting client in step 72.

[0042] Figure 3b is a flowchart of another embodiment in accordance with the present invention, which is substantially similar to that shown in Figure 3a, except that the flowchart represents the viewpoint of a mid-level server, such as server 18 in Figure 1, which receives a request for DNS information from a lower-level server, such as server 22 in Figure 1. The mid-level server preferably forwards the request for DNS information to an upper-level server, such as server 14 or the authoritative server 12 shown in Figure 1.

[0043] In step 86, the mid-level server preferably receives the DNS information request from the lower-level server with the allowable usage limit  $N_L$  that the mid-

level server previously granted to the lower-level server in response to the request from the lower-level server for the DNS information, a lower bound of the observed usage associated with the DNS information reported by the lower-level server  $O_{L,low}$ , and an upper bound of the observed usage associated with the DNS information reported by the lower-level server  $O_{L,high}$ . If, in step 88, the requested DNS information is not in the cache of the mid-level server, the value of the mid-level server variables corresponding to each of these variables is set to zero in step 90. The mid-level server then sends a request to the upper-level or authoritative server in step 92, which includes the current value of  $N_M$ ,  $O_{M,low}$ , and  $O_{M,high}$ .

[0044] In step 94, the mid-level server preferably receives the requested information with the TTL and allowable usage limit  $N$  from the upper-level or authoritative server. The mid-level server then initializes its residual usage counter to  $N$ , initializes the upper bound of observed usage reported by the mid-level server  $O_{high}$  to zero, and initializes the lower bound of observed usage reported by the mid-level server at the lower-level server  $O_{low}$  to zero in step 96. The DNS information is stored in local cache in the mid-level server in step 98 with the allowable usage limit  $N$ .

[0045] In step 100, an allowable usage limit for the DNS information response to the lower-level server  $N_{resp}$  is chosen as less than or equal to the residual usage counter. The value of  $N_{resp}$  can be chosen based on such factors as the remaining TTL of the DNS response, the number of requesters, which may include clients and lower-level DNS servers, to which the requesting DNS server may subsequently provide the requested information, past history of usage by this and other requesting DNS servers, etc. The lower bound for the observed usage at the mid-level server  $O_{low}$  is incremented by one, the upper bound for the observed usage at the mid-level server  $O_{high}$  is incremented by the allowable usage limit for the DNS information response  $N_{resp}$ , and the residual usage counter is decremented by the allowable usage limit for the DNS information response  $N_{resp}$  in step 102.

[0046] Thus, the lower bound for the observed usage at the mid-level server  $O_{low}$  now indicates that the DNS information has been provided to at least one requestor, the lower-level server. The upper bound for the observed usage at the mid-level server  $O_{high}$  now indicates that the maximum number of requestors the DNS

information can be provided to is equal to the allowable usage limit  $N_{resp}$  granted to the lower-level server. The residual usage counter now indicates the remaining usage that may be granted to additional requestors. The requested DNS information is then sent to the lower-level server with the allowable usage limit for the DNS information response  $N_{resp}$  in step 104.

[0047] If the requested DNS information is in cache in step 88 and either the cached information has expired in step 106 or the residual usage counter is zero in step 108, the mid-level server variables are initialized in step 110. Specifically,  $N_M$  is initialized to  $N$ ,  $O_{M,low}$  is initialized to  $O_{low}$ , and  $O_{M,high}$  is initialized to  $O_{high}$ . If the requested DNS information is in cache in step 88, the cached information has not expired in step 106, and the residual usage counter is not zero in step 108, the residual usage counter and upper and lower bounds associated with the lower-level server are set as follows in step 112:

$$[0048] \quad O_{low} := O_{low} - 1 + O_{L,low} \quad (1);$$

$$[0049] \quad O_{high} := O_{high} - N_L + O_{L,high} \quad (2); \text{ and}$$

$$[0050] \quad RUC := RUC + N_L - O_{L,high} \quad (3).$$

[0051]  $O_{low}$  is decremented by (1) in equation 1,  $O_{high}$  is decremented by  $N_L$  in equation (2), and  $N_L$  is added to RUC in equation (3) since corresponding values (1,  $N_{resp}$ ,  $N_{resp}$ ) were applied by the mid-level server to  $O_{low}$ ,  $O_{high}$ , and RUC as presumed values for the lower-level server in step 102. In step 112, the lower-level server is reporting observed values, which are being substituted for the values presumed in step 102. The method then preferably continues with step 110 as described above.

[0052] In accordance with the present invention, the use of DNS information is preferably limited by the number of requestors to which the information has been provided (allowable usage limit). In addition, a request for revalidation of DNS

information provides information concerning the number of requestors to which the DNS information has been provided (observed usage count).

[0053] Several embodiments of the present invention are specifically illustrated and/or described herein. However, it will be appreciated that modifications and variations of the present invention are covered by the above teachings and within the purview of the appended claims without departing from the spirit and intended scope of the invention.